Institute for
Naval Oceanography

TM-3
OCTOBER 1991

SOFTWARE REQUIREMENTS DOCUMENT FOR THE netCDF INTERFACE WITH ECMOP

Version 1.0

LOUISE PERKINS
INSTITUTE FOR NAVAL OCEANOGRAPHY

and

WEN QIAN
INSTITUTE FOR NAVAL OCEANOGRAPHY

Approved for public release; distribution is unlimited. Institute for Naval Oceanography, Stennis Space Center, MS 39529-5005

Institute for
Naval Oceanography

TM-3
OCTOBER 1991

SOFTWARE REQUIREMENTS DOCUMENT FOR THE netCDF INTERFACE WITHIN ECMOP

VERSION 1.0

LOUISE PERKINS
INSTITUTE FOR NAVAL OCEANOGRAPHYY

and

WEN QIAN
INSTITUTE FOR NAVAL OCEANOGRAPHY

Software Requirements Document for the netCDF Interfaces Within ECMOP

Version 1.0 *

A. Louise Perkins

and

Wen Qian

October 22, 1991

Abstract

This document presents the software requirements for the Network Common Data Format (netCDF) interfaces that are being used to modularize the Experimental Center for Mesoscale Ocean Prediction (ECMOP) being built at the Institute for Naval Oceanography (INO). ECMOP is currently under development. It has been designed to be a bread-board test bed for ocean model evaluation and comparisons.

*Institute For Naval Oceanography

Contents

1 Introduction

The goal of this document is to define the netCDF environment within which the Experimental Center for Mesoscale Ocean Prediction (ECMOP), which is being developed at the Institute for Naval Oceanography (INO), will operate. Towards that end we present a data dictionary defining each data item that might appear in any of our netCDF documents.

We will consider only the numerical ocean prediction models SPEM [1] and PRIDER [2][4], and the data assimilation module ASSM [6], in this version of the document. It is expected that the document will be appended as other models are added to ECMOP. There are separate sections for both SPEM and PRIDER, and each new model will need to provide a section describing its own netCDF variable name mapping, also.

We suggest another use for this document, however. New code for the models can adopt either this exact data dictionary or a logical reduction of it. In such a situation the problems of reading a new code would be eased for people familiar with the INO data dictionary for ocean data items.

The netCDF interface has been designed to allow separate distinct task environments to communicate through a "format independent" common data format (CDF). The "net" indicates that these files can be read on any computer and across networks.

It seems appropriate to justify the need for this document. The netCDF file format *is* format independent, and carries its variable names with it. Reading and understanding the file contents then depends on the variable names used and the dimension names that define the shape of those variables. By coupling a data dictionary with a netCDF file, the contents of that file are easier to understand and use through a simple interface, without much assistance. This is our goal here.

2 Information Description

A description of the netCDF can be found in [7]. INO's purpose in moving to netCDF is to provide a more flexible environment for the 1992 assessment. At that assessment we will be

testing many aspects of each model. *After* the assessment we need to be able to combine the more successful aspects of each model into a composite model which, hopefully, is greater than the sum of its parts. We anticipate that the netCDF generality will make any redesign work proceed much quicker.

The netCDF software comes with two utilities, *ncgen* and *ncdump*. These can be used, in a limited sense, instead of a universal interpretor for the CDF. Given a file description in CDL, *ncgen* will generate a set of C or FORTRAN subroutine calls that write the appropriate information to a netCDF file. The INO written utility *inoconv* (Sec. 8.2) will change the "write" subroutine calls to "read" subroutine calls. This utility produces a corresponding set of subroutine calls that read the netCDF file.

There is a one-to-one mapping between the netCDF read and write subroutine calls, and fortunately the parameters are the same for each corresponding call. The utility maps

```
ncvpt1  ->  ncvgt1
ncvp1c  ->  ncvg1c
ncvpt   ->  ncvgt
```

But other subroutine calls must be ignored. The CDF structure provides for a define mode or state. In define state the attributes and dimensions are established. This needs to be done only for the initial write routine. Also there are options that rename attributes within the file headers, and synchronize (checkpoint) the current status of the write program with the file contents. We do not allow any of these subroutines to be called in the read subroutine interface to the netCDF file. The *inoconv* utility comments out these calls. The commented out routine calls are listed below.

```
ncredf   - enter define mode
ncendf   - exit define mode

ncsnc    - force a checkpoint to the file

nccre    - create a netCDF file
ncddff   - create a dimension
ncapt    - create an attribute
ncaptc   - create an attribute

ncadel   - delete an attribute

ncaren   - rename an attribute
```

Two subroutines are then available to read and write the netCDF file and these subroutines can be linked and then loaded along with the routines that wish to access this data. Each model then needs only one set of interface routines, instead of a separate set of routines for each file format.

Of course the routine must be regenerated each time a change has to be made to the file format, but the utility usage is straightforward [7]. After regenerating the netCDF calls using the provided utilities, they must be relinked into the routine using them.

In the future we plan to add the capability of reading and writing to and from a buffer instead of a file. This amounts to formatting an internal buffer. This would save unneeded file access time when running on very large supercomputers.

We now present the detailed description of the problem the netCDF interfaces are being used to solve. This includes human-machine interaction, and hardware and software elements.

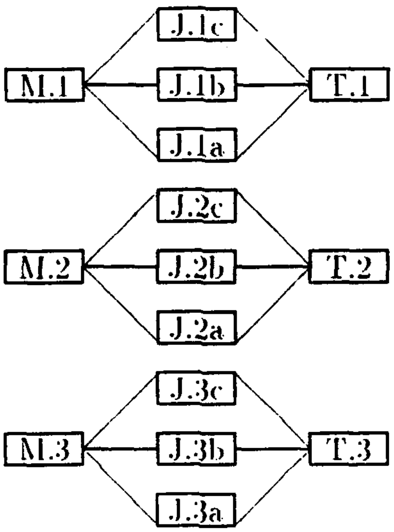

Figure 2.1: Full Connection without netCDF

2.1 Data Flow Diagrams

We are using the common data format to reduce an $\mathcal{O}(\backslash^{\epsilon})$ complexity problem to a problem of $\mathcal{O}(n)$. With separate formats for each file, all tasks must have communication jackets for all other tasks. For example, with three models and three auxiliary general purpose processing tasks, we would need nine jackets to convert between the three varieties of each component, M and T (Fig. 2.1). If we were to add two more varieties of models and tasks the number of jackets required would jump to 25.

You can see that this is a fast growing communications problem. With the netCDF jackets the situation becomes more manageable. With a universal translator the communication paths would be as in Fig. (2.2).

That is, full connectivity can be achieved with an $\mathcal{O}(n)$ linear number of netCDF jackets.

For a system that needs full connectivity real-time the advantages of simple connections far outweigh the cost of the translations.

For the current INO system, ECMOP, such real-time connectivity is not required. This is fortunate because the authors could not find an existing universal translator, and had no ambitions to write their own. Instead we will use the utilities to generate only the specific

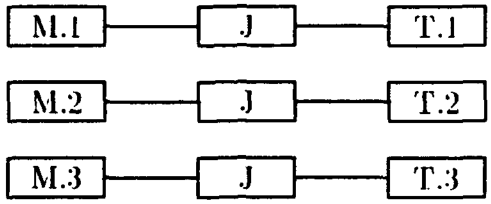

Figure 2.2: Full Connection With netCDF

netCDF jacket needed for the system we are configuring for a test. For each run we will use a connection as in Fig. (2.3)

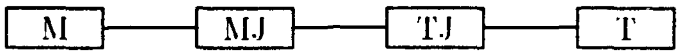

Figure 2.3: Load Time Connections With ECMOP

This nets simplicity of connectivity at run-time without the cost involved for the universal translator. Yet it leaves us upwards compatible so that, if and when a universal translator becomes available we would like to have the option to use it.

2.2 ECMOP Overview

The Institute for Naval Oceanography (INO) was founded in 1985, by a Secretary of the Navy Initiative, to accomplish mesoscale ocean prediction on a global basis and to incorporate university advances in ocean modeling into Naval Oceanography. This document is part of the INO Experimental Center for Mesoscale Ocean Prediction (ECMOP).

ECMOP consists of three major sub-systems: the Verification Module (VERMOD), the Data Sub-System (DASS), and the Visualization Module (VISMOD). The data assimilation

(ASSM) routines described within this document sit between the DASS and the model code. Our test model code, the semi-spectral primitive equation model SPEM, is presented in Hedstrom [1].

A preprocessing utility routine retrieves the necessary data from the DASS through EMPRESS, a Data Base Management System (DBMS), and formats it for ASSM. ASSM then ingests the data real time and transforms it to the requested variables and format needed to feed the model routine. This communication takes place within the netCDF environment [7].

Fig. (2.1) shows a high level diagram of the ECMOP system. The outer box, which is labeled "netCDF Network Environment", serves as a reminder that the subsystem communications will take place within the netCDF environment. These communication links are marked in the diagram with arrows denoting the directional flow of the paths of communication. Rectangular boxes are used to indicate the major software components, man-machine interactions, storage devices, and any form of input or output.

For example, Dr. Louise Perkins, one of the authors of this document, is responsible for this design document, the software characteristics of the ASSM module, the coding, and for the unit testing within the ECMOP system at INO.

Our design philosophy was to design and define in detail the complete functionality expected from the individual modules. But implementation proceeded iteratively, beginning from the simpliest functionality and evolving towards the highest levels.

The DBMS is EMPRESS and is installed on a SUN SPARC station at Stennis Space Center in Mississippi. The Atmospheric Sciences Directorate of the Naval Oceanographic and Atmospheric Research Laboratory (NOARL), formerly NEPRF, has been developing a DBMS system for atmospheric research [3]. INO systems personnel have taken maximum advantage of their expertise and their software, and are using their DBMS design. A preprocessor extracts all data needed for data assimilation for each model run from this DBMS, or from a data file.

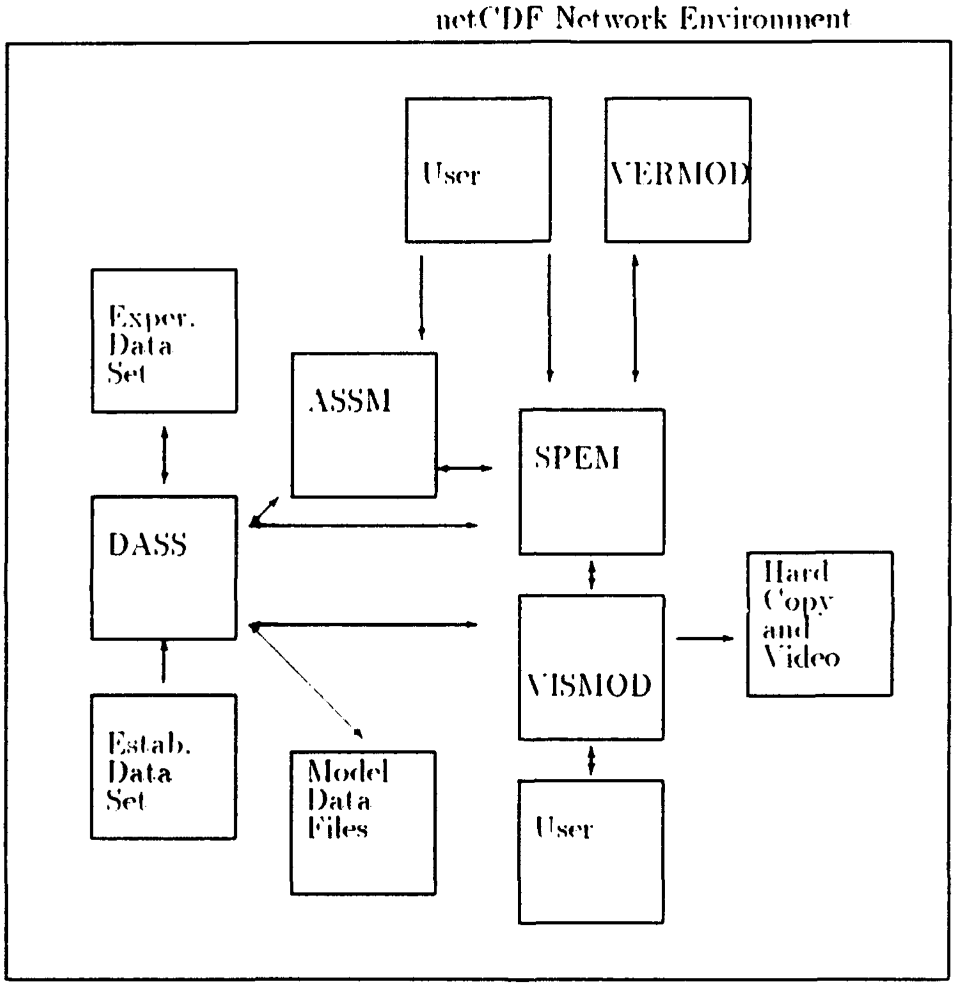

Figure 2.1: SPEM Within ECMOP

Current timing estimates on the Massachusetts Institute of Technology Supercomputer Facility (MITSF) Cray-2 indicate that approximately two Cray minutes are required on one processor for each day the SPEM model is advanced. The Cray-2 at the Massachusetts Institute of Technology (MIT) is not currently heavily utilized so that the CPU time and the wall clock time are comparable. Hence we consider this a real-time requirement for our data access.

The DBMS resides on a separate computer. and network file transport is needed to upload the data onto the larger supercomputer. For this reason we use a preprocessor and assemble the netCDF data files before the actual run begins. The indefiniteness of future data assimilation data needs. coupled with a lack of timing results for the DBMS and the network access path. made this our safest design choice.

Model output can be stored by the DBMS. using a postprocessor. and VERMOD will be able to obtain the output fields and compare one against another. VISMOD is used to display graphical results after or during a test.

3 High Level Design

As an example. we describe the design and development of the ASSM module. The ASSM module is being implemented in four iterative steps (Fig. (3.5)). The first functional assimilation program, the innermost square, is called ASSM HC (Hard Coded). It provides the functionality to incorporate current meter data available at numerical grid points. ASSM HC was a proof-of-concept. coded by Dr. Roberta Young. It was hardcoded without flexibility, and no documentation is available. It is not considered part of ECMOP.

Coding for the ASSM BS (Basic SPEM) was the second iteration and has been used exclusively with the SPEM model.

Next the connection to a preprocessor that retrieves data was implemented. This is the third level of iteration. ASSM DB.

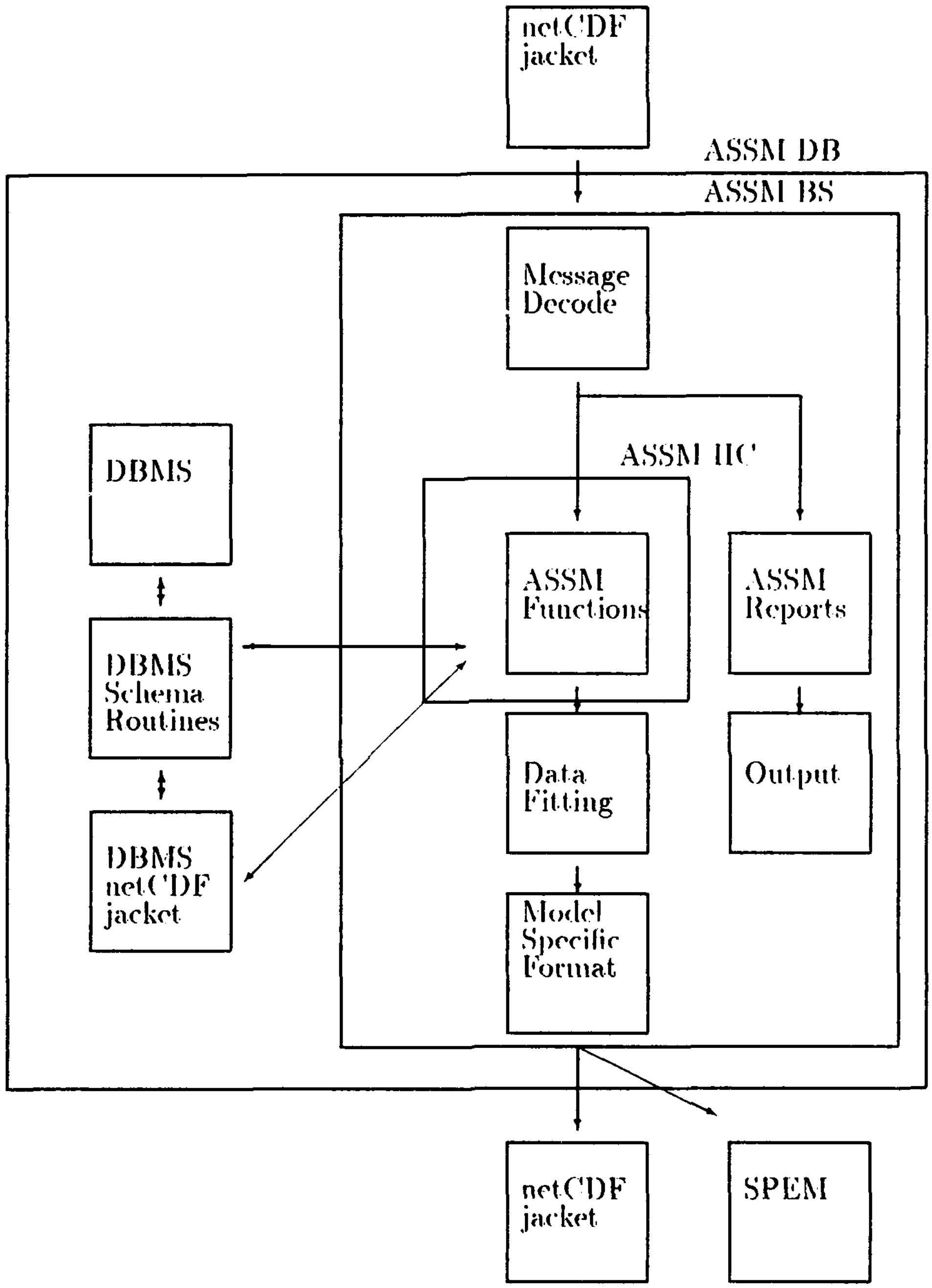

Figure 3.5: ASSM CD

Next the netCDF interfaces for ASSM were implemented. This completed the ASSM utility: the fourth iteration is referred to as ASSM CDF.

A unit test of SPEM within the ECMOP environment will be performed when the ECMOP system is ready. This is expected to locate the interface bugs.

ASSM CDF can be expanded to support multiple models. Additional users are expected to be the Princeton Model (PEDAM), configured for the North Atlantic, and the Data Assimilation Research and Transition (DART) program of NOARL. These groups will enter ECMOP with verification coding and interface standards completed: ASSM will provide an interface for nudging data assimilation into their codes.

Future iterations may be funded for transition to the Naval Oceanography Command. These would be general purpose versions which r quire more rigorous testing and validation than is described here.

4 Data Dictionary

This section presents the data dictionary we are using. All words are abbreviated to four or less letters. Less than four is acceptable only for words less than four letters, or for well established commonly accepted abbreviations. The abbreviations selected followed some general guidelines. The first guideline was to drop all vowels when possible, except when the first letter is a vowel. We rarely drop the first letter in the word unless it is a number or a commonly used English language abbreviation. If we still need more reduction, we next pick the most important consonants to retain. Third a visual inspection for readability was used.

The shorter abbreviations are three or less letters. It is possible to have a hierarchy of shorter abbreviations. We first tried to shorten an abbreviation by removing letters from the end of the long abbreviation. When this was not unique, we retained the dominant sounds of the accented (stressed) syllables [5].

The type field is keyed:

- desc - description
- attr - attribute
- rvar - real variable
- ivar - integer variable
- cvar - character variable
- bvar - bit or logical variable
- gvar - generic variable
- op - operation
- unit - unit of measure

Full_Name	Abbr_Name	Short	Type	Desc
abbreviated	abbr	abr,ab	desc	
absolute	abs	al	op	
access	acc	as	op	
acoustic	acst	ac	attr	
action	actn	atn,an	op	
active	actv	av	desc	
advection	advc	ad,a	rvar	
adjoint	adjt	aj	op	
age	age		attr	
agency	agcy	agc	desc	
air	air	ai	rvar	
altimetry	altm	al	attr	
amount	amt	am	desc	
anemometer	anmt	anm	attr	

angle	ang	ag	op	
anomaly	anom		desc	
ansi	ansi		desc	
area	area	ar	op	
array	arry	ara	desc	
ascending	asc		attr	
assimilation	assm	ass,as	op	
associative	assc	act	op	
atmosphere	atms	af	attr	
attached	att		desc	
attribute	attr	at	attr	
avhrr	avhr		attr	Adv. Very Hi Resol. Raidometer
background	bkgr		attr	
band	band		attr	
bandwidth	bdwh		desc	
baroclinic	bcnc	bc	attr	
barotropic	btpc	bt	attr	
base	base		desc	
beaufort	bfrt		attr	Beaufort wind strength
becomes	bcom		op	
begin	bgn	b	desc	
biology	biol		desc	
bit	bit		desc	
bitstream	bstr		bvar	
bogus	bgus	bgs	desc	
boresight	bors		attr	sensor orientation
bottom	bot	bm	desc	
boundary	bdry	bd	desc	edge or limits

brightness	brgt		rvar	luminance
byte	byte		desc	
calculate	calc		op	
calls	call		op	
callsign	csgn		attr	call-sign of station or ship
calm	calm		desc	
catalog	cat		desc	descriptive list of options
category	catg		desc	division of classification
cell	cell		desc	
celsius	cels	ce	attr	
center	cntr		desc	
change	chg		desc	modification or alteration
channel	chan		attr	
characteristic	chr		desc	
charge	chrg			
cirrus	cirr	ci	desc	type of high cloud
classification	clas		desc	systematic organization
clear	clr		desc	
climatology	clmo	cmo	attr	data pertaining to climate
cloud	cld		desc	
cm	cm		attr	central memory
coast	cst		attr	
code	code	cd	op	an encoding
coefficient	coef	c	gvar	
column	col		desc	
command	cmd		op	order or instruction
comment	cmt		desc	
common	com		desc	

compilation	cmpl		op
complexity	cplx		op
component	cmp		attr
computer	cptr		desc
concentration	conc		attr
condition	cond		desc restricting or modifying factor
conditional	cndl		op
confidence	cnfi		attr degree of certainty
constant	cnst		attr
contains	cntn		op
continuous	cont		desc
contour	cntr		desc isoline
control	ctrl		desc
convective	conv		rvec
convergence	cnvg		op
conversion	cnvn		op transformation between states
copies	copy		op
Coriolis	corl	cor,cl	op
correction	corr		op to meet a standard
count	cnt		attr Numeric value for cardinality
country	ctry		desc
coverage	cvrg		desc extent of coverage
cp	cp		desc central processor
creates	cre		op
creation	crtn		op
criticality	crit		desc
crossection	xsct		op
crossreference	xref		op

cube	cube	cb	desc
current	curr		rvar
curviture	curv		rvar
cycle	cyc		ivar
dampen	damp		op
data	data	d	gvar
date	date	dt	attr time in days
day	day		attr
decay	decy	dcy	
deck	deck		desc of punched cards
decode	dcd		op
deep	deep		rvar
delayed	dlyd		op
density	dens	dn	rvar
depth	dpth	dph,dp	rvar
depression	dprs		rvar a curviture upwards
descending	dsd		op
describes	dscr		desc
description	dsc		desc Textual appendage
destination	dest		desc
device	dev		desc
dewpoint	dwpt		rvar
diameter	dia		rvar
dictionary	dict		desc
difference	dif	df	op
diffusivity	diff	di	op
dimension	dim	dm	desc
direction	dir		desc

directory	drct		desc
disk	dsk		desc
displacement	dspl		rvar
disposition	disp		desc
distance	dis	ds	rvar
distribution	dist		desc
diurnal	drnl		desc
domain	dom		desc limits of variable
Dopler	dplr		desc
east	e		desc
echo	echo		desc
eddy	eddy	ey	desc
effective	eff		desc
elapsed	elps		desc
elevation	elev	elv,ev	rvar
end	end	ed	desc
energy	engy		rvar
executable	xqt	xq	desc
executes	exec		op
expanded	xpnd		desc
external	xtnl	xt	desc
factor	fctr	ftr,fr	attr
fine	fine		desc
first	1		desc
flag	flag	fl	desc boolean flag
float	flot	flt	desc
flux	flux	f	rvar
formal	frml		desc

friction	frct	frc	rvar	
front	frnt		desc	
full	full		desc	
function	func	fun,fc	op	
geographic	geog		desc	taxonomy of surface of earth
geosat	gsat		desc	
get	get	gt	op	
GHz	ghz	gh	attr	
global	glbl		desc	worldwide, comprehensive
gravity	grvt	gvt	rvar	
grid	grid	gr,g	desc	
ground	grnd		desc	surface of earth
gt	gt		op	greater than
guess	gues		op	estimation
heavy	hvy		desc	
heat	heat	he	rvar	
height	ht		rvar	
hertz	hz	h	attr	
high	hi		desc	
horizontal	horz		desc	
hour	hour		rvar	
hydrography	hydr		attr	
identity	id		desc	ident. for object or attribute
image	imge	im	desc	
index	indx	ix	desc	
inactive	inac		desc	
influence	infl	if	desc	
information	info		desc	

infrared	infr		attr	
ingest	ngst		op	
input	iput	ip	op	
insertion	isrt	in	op	
instaneous	inst		desc	
instruction	inst	itr	op	
internal	itnl	it	desc	
interval	itvl	ivl,il,i	desc	
inverted	inv	iv,v	op	
ir	ir		attr	infra-red
is	is		desc	exists
isentropic	isnt		desc	equal potential temperature
isobaric	isbr		desc	
kilogram	kilo	kg	unit	
kinematic	knmt		desc	
Lagrangian	lagn		attr	
land	land	lnd	desc	
last	last		desc	
latent	ltnt		desc	
latitude	lat		rvar	
left	left	l	desc	
length	lnth		gvar	
level	lvl	lv	desc	
light	lit		desc	
list	list		desc	any list of words or numerals
llt	llt		attr	info about llt data
load	load		desc	
location	lctn		desc	

logical	logc	lg	desc	
long	long			
longitude	lon		rvar	
low	low	lw	desc	
lt	lt		op	less than
lu	lu		desc	logical unit
maps	maps	mps	op	
marine	marn		desc	
mask	mask	msk,ms	op	
mass	mass		rvar	
matches	mtch		op	
maximum	max		op	
mean	mean	mn	op	
meander	mand		desc	
method	mthd	mt	attr	
meters	mtrs	mtr,mr	unit	
middle	mid	m	desc	
millibar	mb		attr	
minimum	min		op	
minutes	mnts		attr	
missing	miss		desc	
mixed	mixd		desc	
model	modl	mdl	desc	
modifies	mdfy		op	
molecular	mlcr		attr	
momentum	mmtm	mm	rvar	
moored	mord		attr	
nadir	ndr		desc	point below surface observation

name	name		desc literal value of object
next	next		desc
north	n		desc 0 degrees
northeast	ne		desc 45 degrees
northwest	nw		desc 315 degrees
not	not		op
nudging	nudg	ndg,ng	op
number	num	nu	ivar index into series of objects
numeric	nmrc	nmc	desc
observation	ob	o	desc
occupies	occp		desc
ocean	ocn		desc
oe	oe		attr info about orbital element
operational	oper		desc
operator	op		attr operation to perform
ordinal	ord		desc
output	oput	op	op
particle	prtc	pc	desc
past	past		desc
path	path	pth,ph	desc
per	per	pe	op
period	perd		desc duration of interval event
permanent	perm		desc
pk	pk		desc acronym - primary key
point	pt		desc
points	pts	p	desc
precipitable	prcl		desc
present	prst		desc

pressure	pres		rvar
prevailing	prvl	pr	desc
previous	prev	pv	desc
primary	prim		desc
print	prnt	prt	op
probability	prbl		op
profiles	prof	prf,pf	desc
put	put		op
q	q		desc terrain resolution
qualitative	qltv		desc
radiation	rdtn		rvar
radius	rad	ra	rvar
Rafos	rafs	raf	desc
ramp	ramp		op
random	rndm		desc
range	rang		desc
rate	rate	rat	desc
read	read	rd	op
references	ref		desc
relative	rltv		desc
remark	rmrk		desc addendum for clarity
report	rpt		desc
request	rqst	rq	op
requester	reqr		desc
requires	req		desc
research	rsch		desc
resembles	rsmb		desc
resolution	rslt		desc

responsible	resp		desc
return	rtn	rt	op
right	rht	r	desc
Rossby	rsby		desc
salinity	salt	sal,sl	rvar
satellite	sat	st	attr
save	save	sv	op
scale	scl	sc	op
sea	sea	se	desc
second	2		desc
secondary	scdy		desc
seconds	sec		desc
sensible	snsb		desc
sensor	sens		desc
shallow	shlw		desc
shear	shr		rvar
ship	ship		desc
sigma	sgma		rvar
significant	sig		desc
simple	smpl	sp	desc
size	size	sz	attr
smooth	smth		desc
sounding	sndg		desc
solar	sol		desc
sonic	snc		desc
sounder	sndr		desc
south	s		desc 180 degrees
southeast	se		desc 135 degrees

southwest	sw		desc 225 degrees
space	spc		desc
spawns	spwn		op
speed	spd		rvar
squared	sqrd	sq	op
start	strt		desc
station	stn		desc
step	step		desc
stream	strm		desc
stress	strs	str	rvar
subset	sbst	ss	desc
subjective	sbj		desc
subordinate	sub		desc
sum	sum		op
superior	supr		desc
surface	surf	sf	desc
Sverdrup	sver		attr
temperature	temp	tp	rvar
temporary	tmp		desc
terrain	terr		desc
text	text	txt	desc qualitative info. in text
thermocline	thcl		desc
third	3		desc
threshold	thld		desc
time	tm	t	desc absolute time
topography	topo		desc
total	totl	ttl	desc
track	trck	tk	desc

transformation	trfm		op	
transient	trns		desc	
transport	tprt		desc	
turbulence	trbl	trb,tl	desc	
type	type	ty	attr	selection from a category
u	v1dm		rvar	velocity in first dimension
ucomponent	ucmp		desc	
unexpanded	unxp		desc	
unit	unit	u	desc	
units	unts	us	desc	
update	updt	ud	op	
upper	uppr	up	desc	
urban	urbn	urb	desc	
uses	use		desc	
usn	usn		desc	US Navy
v	v2dm		rvar	velocity in second dimension
valid	vald		desc	
value	val		desc	numeric represent. of quantity
variable	vrbl	vb	desc	
variance	var		desc	
vcomponent	vcmp		desc	
velocity	vel	v	rvar	
vertical	vert	vr	desc	
viscosity	visc	vs	rvar	
visible	vis		desc	
vorticity	vort		rvar	
w	v3dm		rvar	velocity in third dimension
warming	wrmg		desc	

water	wtr		desc
wave	wav		desc
wban	wban		desc weather bureau Air Force Navy
weather	wthr		desc
weight	wght	wt	desc
west	w		desc 270 degrees
wetbulb	wetb		attr wetbulb thermometer
whitecap	wcap		desc crest of wave is foam
width	wdth		desc
wind	wind	wnd	rvar
wmo	wmo		desc World Meteoro. Organization
word	word		desc
write	wrt	wr	op
xtom	xtom		desc computation excludes toms
yaw	yaw		op rotation about vertical axis
year	year	yr	desc
zenith	zen		desc vertically above observation
zone	zone		desc
zonal	znl		desc

5 netCDF Jackets

The netCDF jackets are responsible for mapping model variable names to the netCDF long and short abbreviated names given in the data dictionary in Sec. (4). Complex names, consisting of underscore-concatenated long form abbreviations, are used for the netCDF file variable names, whenever possible. Short form abbreviations are concatenated without the underscore. In length limited programming languages the shorter abbreviations are used in the netCDF file.

This section includes the currently defined mappings for both the SPEM and Princeton models as well as for the ASSM module. All can be viewed as components in the ECMOP environment.

5.1 SPEM to netCDF Mapping

We map the SPEM variable names, which are listed in the column "model", to the "Short-form" column abbreviations available in the netCDF file because SPEM is a FORTRAN program.

Model	long netCDF	short netCDF	Description
dt	time_step	dt	
dtyp	data_type	dty	
disp	flag_dcd	fldcd	
h	ht	ht	
ipre	flag_prev	flpr	
isave	flag_save	flsv	
iupd	flag_updt	flud	
idat	date	date	
irw	flag_rd_wrt	flrdwr	
irc	flag_rtn_code	flrtcd	
irng	data_iflc	diflc	
irpt	logc_unit_rpt	lurpt	
ixb	arry_size_bgn_1_dim	b1dm	
ixe	arry_size_end_1_dim	ed1dm	
iyb	arry_size_bgn_2_dim	b2dm	
iye	arry_size_end_2_dim	ed2dm	

izb	arry_size_bgn_3_dim	b3dm	
ize	arry_size_end_3_dim	ed3dm	
iuptr	1_dim_vel_ob_lctn_1_dim	v11odm	
ivptr	2_dim_vel_ob_lctn_1_dim	v21odm	
irptr	dens_ob_lctn_1_dim	dn1odm	
juptr	1_dim_vel_ob_lctn_2_dim	v12odm	
jvptr	2_dim_vel_ob_lctn_2_dim	v22odm	
jrptr	dens_ob_lctn_2_dim	dn2odm	
kuptr	1_dim_vel_ob_lctn_3_dim	v13odm	
kvptr	2_dim_vel_ob_lctn_3_dim	v23odm	
krptr	dens_ob_lctn_3_dim	dn3odm	
loca	lctn	lctn	
l	1_dim_grid_size	1dmgr	
m	2_dim_grid_size	2dmgr	
mtppt	max_topo_size	toposz	
numu	1_dim_vel_ob_size	v1odm	
numv	2_dim_vel_ob_size	v2odm	
numr	dens_ob_size	dnodm	
n	3_dim_grid_size	gr3dm	
nptlg	num_pts_lon	nplon	grid_size
nptlt	num_pts_lat	nplat	grid_size
psi	vort	vort	
rampuv	per	per	period to interpolate thru
resol	rslt	rslt	
rho	dens	dens	
rlllt	low_left_lat	lwllat	coverage, geographic_location
rllng	low_left_lon	lwllon	coverage, geographic_location
robs	dens_ob	dno	

ros2	cnst_rsby_rad	rsbyrd	Rossby radius
rclm	dens_clmo	dnclmo	
rdcy	arry_data_iflc	aradif	
rdata	dens_data	dnd	
rudec	1_dim_vel_coef_assm	v1cas	
rvdec	2_dim_vel_coef_assm	v2cas	
rrdec	dens_coef_assm	dncas	
scalf	scl_fctr	sclftr	
ssbathy	sbst_topo	sstopo	
sslllt	sbst_low_left_lat	sslwl2	domain_values or coverage
ssllg	sbst_low_left_lon	sslwl1	domain_values or coverage
ssurlt	sbst_uppr_rht_lat	ssupr2	domain_values or coverage
ssurlg	sbst_uppr_rht_lon	ssupr1	domain_values or coverage
t	curr_tm	currtm	
tobs	ob_tm	obtm	
tdata	elps_tm	elpstm	*time since last data assimilation*
u	1_dim_vel	v1dm	
uclm	1_dim_vel_clmo	v1dmcl	
udata	1_dim_vel_data	v1dmd	
urlat	uppr_rht_lat	uprlat	
urlng	uppr_rht_lon	uprlon	
uobs	1_dim_vel_ob	v1dob	
vobs	2_dim_vel_ob	v2dob	
v	2_dim_vel	v2dm	
vdata	2_dim_vel_data	v2dmd	
vclm	2_dim_vel_clmo	v2dmcl	
w	3_dim_vel	v3dm	

5.2 PRIDER to netCDF Mapping

The mapping for the PRIDER package is given in this subsection. We map the PRIDER variable names, which are listed in the column "Model", to the Shortform abbreviation because PRIDER is a FORTRAN routine.

Model	netCDF	Short	Description
-----	------	-----	-----------
aam2d	eddy_visc	eddyvs	
alat	grid_cntr_lat	grlat	
alon	grid_cntr_lon	grlon	
ang	ang_1_dim_2_dim	ang12	Angle between 1 and 2 dimensions
art	area_1_dim_2_dim_grid	area12	Area defined by cells in 1 and 2 dim
aru	area_cntr_vel_1_dim	v1dm	
arv	area_cntr_vel_2_dim	v2dm	
aam	knmt_visc	knmtvs	Kinematic Viscosity
cbc	coef_bot_frct	cbtfrc	
cor4	corl	corl	
curv42d	sum_corl	sumcrl	
covrhn	grvt_rdtn_n_bdry	gvtnbd	Gravity radiation at northern boundary
covrhe	grvt_rdtn_e_bdry	gvtebd	
d	ht	ht	
dtef	vert_prof	vertpf	
dum	1_dim_vel_mask	v1dmms	
dvm	2_dim_vel_mask	v2dmms	
dx	1_dim_dis_invl	ds1dm	
dy	2_dim_dis_invl	ds2dm	
dt	3_dim_dis_time_step	ds3dmt	

dte	xtnl_time_step_sec	xtnlt
dti	itnl_time_step_sec	itnlt
dz	3_dim_dis_invl	ds3dmi
dzz	3_dim_dis_mid	ds3dmm
dzr	3_dim_dis_inv_invl	ds3dmv
egf	elev	elev
egb	elev_prev_time_step	elevts
el	elev_xtnl	elevxt
elb	elev_prev_time_step_xtnl	elevpv
et	elev_int	elvint
etb	elev_prev_time_step_itnl	elvitt
etf	elev_next_time_step_itnl	elvit
eln	sea_lvl_n_bdry	selvnb
ele	sea_lvl_e_bdry	selveb
els	sea_lvl_s_bdry	selvsb
fluxua	advc_1_dim	a1dm
fluxva	advc_2_dim	a2dm
fsm	sea_lvl_mask	selvms
ftc	damp_temp_salt_fctr	tpsal
grav	grvt	grvt
h	3_dim_depth	dph3dm
i	1_dim_grid_indx	g1dmix
im	1_dim_grid_size	g1dmsz
int	ixtl_time_step	ixtlt
iprint	prnt_invl	prtiv
jm	2_dim_grid_size	g2dmsz
j	2_dim_grid_indx	g2dmix
k	3_dim_grid_indx	g3dmix

kh	vert_diff_coef	vrdic
km	vert_knmt	vrknmt
kq	vert_trbl	vrtrbl
l	trbl_lnth_scl	trbl
mode	type_calc	calcty
nts	next_time_step	nextt
pts	prev_time_step	prevt
psi	strm_func	strmfc
q2	2_trbl_engy	trbl2
q2b	2_trbl_engy_pts	trb2pt
q2l	2_trbl_engy_lnth_scl	trb2sc
q2lb	2_trbl_engy_lnth_scl_pts	tlscpt
rho	dens	dens
rhof	dens_pts	denspt
rmean	area_mean	areamn
ramp	ramp_strt_fctr	rampfr
s	salt	salt
sb	salt_pts	saltpt
sbe	salt_e_bdry	salebd
sbn	salt_n_bdry	salnbd
sbs	salt_s_bdry	salsbd
sbw	salt_w_bdry	salwbd
smean	mean_salt	mnsalt
t	temp_cels	tempce
tb	temp_cels_pts	tpcept
tmean	temp_cels_mean	tpcemn
tprnu	wght_fctr	wtfctr

time	time	time	
tbe	temp_e_bdry	tpebd	
tbn	temp_n_bdry	tpnbd	
tbs	temp_s_bdry	tpsbd	
tbw	temp_w_bdry	tpwbd	
uabe	btrp_vel_e_bdry	btvebd	
u	1_dim_vel	v1dm	
uf	1_dim_vel_pts	v1dmpt	
ub	1_dim_vel_num_time_step	v1dmnu	
ua	1_dim_vel	v1dm	
uab	1_dim_vel_pts	v1dmpt	
uaf	1_dim_vel_num_time_step	v1tm	
umol	mlcr_diff	mlcrdi	
va	2_dim_vel	va2dm	
v	2_dim_vel	v2dm	
vf	2_dim_vel_pts	v2dmpt	
vb	2_dim_vel_num_time_step	v2dmnu	
vh	vert_prof	vertpf	
vhp	vert_prof_pts	vrpfpt	
vabn	btrp_vel_n_bdry	btvnbd	
vabs	btrp_vel_s_bdry	btvsbd	
vab	2_dim_vel_pts	v2dmpt	
vaf	2_dim_vel_num_time_step	v2dmnu	
wssurf	salt_flux_surf	salfsf	
wtsurf	heat_flux_surf	htfsf	
wubot	1_dim_mmtm_flux_bot	mm1bt	Momentum flux at bottom in 1 dimension
wvbot	2_dim_mmtm_flux_bot	mm2bt	
wusurf	1_dim_mmtm_flux_surf	mm1sf	

wvsurf	2_dim_mmtm_flux_surf	mm2sf
w	3_dim_vel	v3dm
z	3_dim_dis	ds3dm
zz	3_dim_dis_mid_invl	ds3iv

5.3 ASSM Routine

The mapping for the ASSM package is given in this subsection. We map the ASSM variable names, which are listed in the column "Model", to the Shortform abbreviation because ASSM is a FORTRAN routine.

Model	netCDF Long Name	netCDF Shortform	Description
-----	---------	---------	-----------
asfun	assm_func	asfun	what to do
biold	biol_data	biold	
clm	clmo_arra	cmara	
data	data	data	
dens	dens	dens	
dt	time_step_size	tstpsz	
gridpt	grid_pt	gridpt	
idat	curr_date	currdt	
infogt	info_get	infogt	
inst	inst	inst	
io	iput_oput_flag	ipopfl	
ipre	prev_flag	prevfl	
irc	rtn_code_flag	rtncd	
irng	rang_iflc	rangif	

is	intr	intr	
isave	save_flag	savefl	
iupd	updt_flag	updtfl	
l	size_1_dim	sz1dm	
lagnd	lagn_flot_info	lagnd	
lctn	lctn	lctn	
m	size_2_dim	sz2dm	
mdlasd	modl_vrbl_assm_data	mdlasd	
mdldsc	modl_grid_dsc	mdldsc	what it is
mdlvb	modl_vrbl_val	mdlvr	
mean	mean	mean	
model	modl	mdl	who to assimilate for
n	size_3_dim	sz3dm	
nudclm	nudg_clmo	nudgcm	
nump	num_pts	nupts	
ob	ob	ob	
pres	pres	pres	
rdcy	rate_decy	ratdcy	
rho	dens_arra	dens	
spce	spce	spce	
sver	tprt_sver	sver	
t	time	t	
temp	temp	temp	
time	time	time	
tobs	time_ob	tob	
u	vel_1_dim_arra	v1dm	
v	vel_2_dim_arra	v2dm	
vel	vel	vel	

```
w       vel_3_dim_arra          v3dm
wndstr  wnd_strs                wndstr
```

5.3.1 Data Space

There are many "standard" data structures in data space. The best structure for each data type with respect to storage needs, retrieval and update needs are still open issues. In this section, with an eye towards the future, we set down the contents of groups of data by collection instruments or commonly associated data types. The instrument is listed and then some of the data types we might expect to be associated with that instrument are listed under it.

```
ADP:
   acst_dplr_prof (adp)
   vel_prof_vert (vprfvr)

Gross Feature Locations (Bogus Maps):
    mndr
    eddy
    frnt_trak (frnttk)
    axis
    n_wall

Hydrography:
  Hydr
    sea_surf_ht (sesfht)
    sea_surf_temp  (sesftp)
```

```
Inverted Echo Sounders:
  inv_echo_sndr (ies)
    thcl_dpth

Path Characteristics:
  path_char  (pthchr)
    dspl
    ang_prtc_path  (agpcph)
    curvature (curv)

Satellite Altimetry:
  sat_altm (satal)
    sea_surf_ht  (sesfht)
    infr (infr)
```

6 Constant Values

For actual numbers, the value can be concatenated with the unit or type. For example $10 - minutes$, or the associated short form $10mnts$, can be used for a constant time value of ten minutes.

7 Constraints on Data Items

Some words are not allowed because other words are equivalent and "better" in some sense. The following words are *not* to be added to the data dictionary in Sec. (2):

```
altitude - use height or elevation
average    use mean
```

```
datetime    use time
definition use description
increment   use interval
mode        use description or type
quantity    use amount
```

8 Functional Description

This section describes the procedural details of using netCDF within ECMOP. There are several options. Binding the netCDF interface to the model can be made at compile time by generating new subroutines for any changes. Binding can occur at load time by selecting one of several subroutines to use, and binding can occur at run time by making dynamic decisions about the contents of a netCDF file. For the FY92 time frame, we suggest load time binding using makefiles within the UNIX environment. This is the most cost effective in the short run.

8.1 Processing Narrative

This section will walk us through the algorithms available for use within each utility and each subroutine.

The netCDF provides several data formats:

- byte
- character
- short
- long (32 bit)
- float (32 bit IEEE floating-point)
- double (64 bit)

The INO ECMOP environment has defined some common attributes:

- units
- long-name
- valid-range
- valid-min
- valid-max
- scale-factor
- add-offset
- missing-value
- C-format
- FORTRAN-format
- title
- history

Given the contents of data files that exist or that need to be created, the user must generate a header file that describes the contents of the file. The SPEM netCDF history file interface is written with the following FORTRAN program. The corresponding CDL file is presented after the program.

```
***************************************************
*        Name:  MCSSTgencdf                       *
* Programmer:  Phuong Van Leach                   *
*    Company:  Institute for Naval Oceanography   *
*       Date:  July 19, 1991                      *
***************************************************
*                                                 *
*     This FORTRAN program creates a netCDF file *
*  in which the scientific data can be accessed  *
*  and shared in a form that is self-describing  *
*  and network-transparent.                       *
*  *
***************************************************

      program hist


* links to netCDF library
      include 'netcdf.inc'

**********************************

* declares variables for parameter
      integer nhist
      integer charlen
      integer num_long
      integer num_lat
      integer num_levels
```

```
* defines maximum dimensions
      parameter (nhist = 4)
      parameter (charlen = 36)
      parameter (num_long = 130)
      parameter (num_lat = 130)
      parameter (num_levels = 5)

* declares hypercube
      integer strt2D(2), cnt2D(2)
      integer strt3D(3), cnt3D(3)

* declares hypercube data
      data strt2D /1, 1/
      data strt3D /1, 1, 1/
      data cnt2D /num_long, num_lat/
      data cnt3D /num_long, num_lat, num_levels/

* declares return error code
      integer iret

* declares netCDF id
      integer cdfid

* delcares dimension ids
      integer nhistdim, charlendim
      integer lpdim, mpdim, npdim, mdim
```

```
* declares variable ids
      integer xiuid,xivid,xitid,xisid,psiid,identid
      integer xlid,elid,hid,fid,uvnu2id
      integer snu2id,tnu2id,uvnu4id,snu4id,tnu4id,rho0id
      integer gid,rdrgid,hminid,hmaxid,rfacid,rdmpid,pmid
      integer pnid,dndxid,dmdeid,dhdxid,dhdeid,dxid,deid
      integer xpid,ypid,piid,sigid,cpid,cfid,cdid,cdzid
      integer cintid,csumid,iprmid,fprmid,mgoptid
      integer xminid,yminid,xmaxid,ymaxid
      integer sclmid,tclmid,uclmid,vclmid
      integer dtid,timeid

* declares variable shapes
      integer dim1(1)
      integer dim2(2)
      integer dim3(3)

*************************************************

* declares data variables
      real xiu(num_long,num_lat,num_levels)
      real xiv(num_long,num_lat,num_levels)
      real xit(num_long,num_lat,num_levels)
      real xis(num_long,num_lat,num_levels)
      real psi(num_long,num_lat)
      character ident(charlen)
      real xl
      real el
```

```
real h(num_long,num_lat)
real f(num_long,num_lat)
real uvnu2
real snu2
real tnu2
real uvnu4
real snu4
real tnu4
real rho0
real g
real rdrg
real hmin
real hmax
real rfac
real rdmp
real pm
real pn
real dndx
real dmde
real dhdx
real dhde
real dx
real de
real xp
real yp
real pi
real sig
real cp
```

```
      real cf
      real cd
      real cdz
      real cint
      real csum
      integer iprm
      integer fprm
      integer mgopt
      real xmin
      real ymin
      real xmax
      real ymax
      real sclm
      real tclm
      real uclm
      real vclm
      real dt
      real time

*****************************************************************
* This routine generates a netcdf file from a data file.       *
*****************************************************************

* creates netcdf file/ enters define mode
      cdfid = nccre('/epoch/data3/leach/hist.cdf', ncclob, iret)

* defines dimensions
```

```
      nhistim = ncddef(cdfid, 'nhist', nhist, iret)
      charlendim = ncddef(cdfid, 'charlen', charlen, iret)
      lpdim = ncddef(cdfid, 'num_long', num_long, iret)
      mpdim = ncddef(cdfid, 'num_lat', num_lat, iret)
      npdim = ncddef(cdfid, 'num_levels', num_levels, iret)

* defines dbms variables
      dim3(3) = npdim
      dim3(2) = mpdim
      dim3(1) = lpdim
      xiuid = ncvdef(cdfid,'xiu',NCFLOAT,3,dim3,iret)
      xivid = ncvdef(cdfid,'xiv',NCFLOAT,3,dim3,iret)
      xitid = ncvdef(cdfid,'xit',NCFLOAT,3,dim3,iret)
      xisid = ncvdef(cdfid,'xis',NCFLOAT,3,dim3,iret)

      dim2(2) = mpdim
      dim2(1) = lpdim
      psiid = ncvdef(cdfid,'psi',NCFLOAT,2,dim2,iret)

      dim1(1) = charlendim
      identid = ncvdef(cdfid,'ident',NCCHAR,1,dim1,iret)

      xlid = ncvdef(cdfid,'xl',NCFLOAT,0,0,iret)
      elid = ncvdef(cdfid,'el',NCFLOAT,0,0,iret)


      dim2(2) = mpdim
      dim2(1) = lpdim
```

```
hid = ncvdef(cdfid,'h',NCFLOAT,2,dim2,iret)
fid = ncvdef(cdfid,'f',NCFLOAT,2,dim2,iret)

uvnu2id = ncvdef(cdfid,'uvnu2',NCFLOAT,0,0,iret)
snu2id = ncvdef(cdfid,'snu2',NCFLOAT,0,0,iret)
tnu2id = ncvdef(cdfid,'tnu2',NCFLOAT,0,0,iret)
uvnu4id = ncvdef(cdfid,'uvnu4',NCFLOAT,0,0,iret)
snu4id = ncvdef(cdfid,'snu4',NCFLOAT,0,0,iret)
tnu4id = ncvdef(cdfid,'tnu4',NCFLOAT,0,0,iret)
rho0id = ncvdef(cdfid,'rho0',NCFLOAT,0,0,iret)
gid = ncvdef(cdfid,'g',NCFLOAT,0,0,iret)
rdrgid = ncvdef(cdfid,'rdrg',NCFLOAT,0,0,iret)
hminid = ncvdef(cdfid,'hmin',NCFLOAT,0,0,iret)
hmaxid = ncvdef(cdfid,'hmax',NCFLOAT,0,0,iret)
rfacid = ncvdef(cdfid,'rfac',NCFLOAT,0,0,iret)
rdmpid = ncvdef(cdfid,'rdmp',NCFLOAT,0,0,iret)
pmid = ncvdef(cdfid,'pm',NCFLOAT,0,0,iret)
pnid = ncvdef(cdfid,'pn',NCFLOAT,0,0,iret)
dndxid = ncvdef(cdfid,'dndx',NCFLOAT,0,0,iret)
dmdeid = ncvdef(cdfid,'dmde',NCFLOAT,0,0,iret)
dhdxid = ncvdef(cdfid,'dhdx',NCFLOAT,0,0,iret)
dhdeid = ncvdef(cdfid,'dhde',NCFLOAT,0,0,iret)
dxid = ncvdef(cdfid,'dx',NCFLOAT,0,0,iret)
deid = ncvdef(cdfid,'de',NCFLOAT,0,0,iret)
xpid = ncvdef(cdfid,'xp',NCFLOAT,0,0,iret)
ypid = ncvdef(cdfid,'yp',NCFLOAT,0,0,iret)
piid = ncvdef(cdfid,'pi',NCFLOAT,0,0,iret)
sigid = ncvdef(cdfid,'sig',NCFLOAT,0,0,iret)
```

```
      cpid = ncvdef(cdfid,'cp',NCFLOAT,0,0,iret)
      cfid = ncvdef(cdfid,'cf',NCFLOAT,0,0,iret)
      cdid = ncvdef(cdfid,'cd',NCFLOAT,0,0,iret)
      cdzid = ncvdef(cdfid,'cdz',NCFLOAT,0,0,iret)
      cintid = ncvdef(cdfid,'cint',NCFLOAT,0,0,iret)
      csumid = ncvdef(cdfid,'csum',NCFLOAT,0,0,iret)
      iprmid = ncvdef(cdfid,'iprm',NCLONG,0,0,iret)
      fprmid = ncvdef(cdfid,'fprm',NCLONG,0,0,iret)
      mgoptid = ncvdef(cdfid,'mgopt',NCLONG,0,0,iret)
      xminid = ncvdef(cdfid,'xmin',NCFLOAT,0,0,iret)
      yminid = ncvdef(cdfid,'ymin',NCFLOAT,0,0,iret)
      xmaxid = ncvdef(cdfid,'xmax',NCFLOAT,0,0,iret)
      ymaxid = ncvdef(cdfid,'ymax',NCFLOAT,0,0,iret)
      sclmid = ncvdef(cdfid,'sclm',NCFLOAT,0,0,iret)
      tclmid = ncvdef(cdfid,'tclm',NCFLOAT,0,0,iret)
      uclmid = ncvdef(cdfid,'uclm',NCFLOAT,0,0,iret)
      vclmid = ncvdef(cdfid,'vclm',NCFLOAT,0,0,iret)
      dtid = ncvdef(cdfid,'dt',NCFLOAT,0,0,iret)
      timeid = ncvdef(cdfid,'time',NCFLOAT,0,0,iret)

* assigns attributes to dbms variables
      call ncaptc(cdfid,xiuid,'long_name',NCCHAR,31,
     1 'velocity in longitude dimension', iret)
      call ncaptc(cdfid,xivid,'long_name',NCCHAR,30,
     1 'velocity in latitude dimension', iret)
      call ncaptc(cdfid,xitid,'long_name',NCCHAR,11,
     1 'temperature', iret)
      call ncaptc(cdfid,xisid,'long_name',NCCHAR,11,
```

```
1 'temperature', iret)
 call ncaptc(cdfid,psiid,'long_name',NCCHAR,36,
1 'horizontal transport stream function', iret)
 call ncaptc(cdfid,identid,'long_name',NCCHAR,40,
1 'run identification (YYMMDD,type of data)', iret)
 call ncaptc(cdfid,xlid,'long_name',NCCHAR,25,
1 'longitude interval length', iret)
 call ncaptc(cdfid,elid,'long_name',NCCHAR,24,
1 'latitude interval length', iret)
 call ncaptc(cdfid,hid,'long_name',NCCHAR,18,
1 'bottom depth model', iret)
 call ncaptc(cdfid,fid,'long_name',NCCHAR,15,
1 'coriollis force', iret)
 call ncaptc(cdfid,uvnu2id,'long_name',NCCHAR,42,
1 'laPlacian coefficient of friction velocity', iret)
 call ncaptc(cdfid,snu2id,'long_name',NCCHAR,42,
1 'laPlacian coefficient of friction salinity', iret)
 call ncaptc(cdfid,tnu2id,'long_name',NCCHAR,45,
1 'laPlacian coefficient of friction temperature',
1 iret)
 call ncaptc(cdfid,uvnu4id,'long_name',NCCHAR,43,
1 'biharmonic coefficient of friction velocity',
1 iret)
 call ncaptc(cdfid,snu4id,'long_name',NCCHAR,43,
1 'biharmonic coefficient of friction salinity',
1 iret)
 call ncaptc(cdfid,tnu4id,'long_name',NCCHAR,46,
1 'biharmonic coefficient of friction temperature',
```

```
1 iret)
 call ncaptc(cdfid,rho0id,'long_name',NCCHAR,16,
1 'constant density', iret)
 call ncaptc(cdfid,gid,'long_name',NCCHAR,7,
1 'gravity', iret)
 call ncaptc(cdfid,rdrgid,'long_name',NCCHAR,23,
1 'bottom drag coefficient', iret)
 call ncaptc(cdfid,hminid,'long_name',NCCHAR,20,
1 'minimum bottom depth', iret)
 call ncaptc(cdfid,hmaxid,'long_name',NCCHAR,20,
1 'maximum bottom depth', iret)
 call ncaptc(cdfid,rfacid,'long_name',NCCHAR,38,
1 'simulation conversion factor (.5 g/p.)', iret)
 call ncaptc(cdfid,rdmpid,'long_name',NCCHAR,27,
1 'weights for nudging forcing', iret)
 call ncaptc(cdfid,pmid,'long_name',NCCHAR,48,
1 'Jacobian of grid transformation in lon dimension',
1 iret)
 call ncaptc(cdfid,pnid,'long_name',NCCHAR,48,
1 'Jacobian of grid transformation in lat dimension',
1  iret)
 call ncaptc(cdfid,dndxid,'long_name',NCCHAR,43,
1 'Hermetian of "pn" with respect to longitude', iret)
 call ncaptc(cdfid,dmdeid,'long_name',NCCHAR,42,
1 'Hermetian of "pm" with respect to latitude', iret)
 call ncaptc(cdfid,dhdxid,'long_name',NCCHAR,50,
1 'derivating of topography with respect to longitude',
1 iret)
```

```
 call ncaptc(cdfid,dhdeid,'long_name',NCCHAR,49,
1 'derivating of topography with respect to latitude',
1 iret)
 call ncaptc(cdfid,dxid,'long_name',NCCHAR,31,
1 'interval in longitude dimension', iret)
 call ncaptc(cdfid,deid,'long_name',NCCHAR,30,
1 'interval in latitude dimension', iret)
 call ncaptc(cdfid,xpid,'long_name',NCCHAR,21,
1 'longitude grid points', iret)
 call ncaptc(cdfid,ypid,'long_name',NCCHAR,20,
1 'latitude grid points', iret)
 call ncaptc(cdfid,piid,'long_name',NCCHAR,2,
1 'pi', iret)
 call ncaptc(cdfid,sigid,'long_name',NCCHAR,39,
1 'stretched coordinate collocation points', iret)
 call ncaptc(cdfid,cpid,'long_name',NCCHAR,20,
1 'polynomial basis set', iret)
 call ncaptc(cdfid,cfid,'long_name',NCCHAR,48,
1 'matrix of basis functions at the sigma locations',
1 iret)
 call ncaptc(cdfid,cdid,'long_name',NCCHAR,44,
1 'derivative of basis with respect to vertical',
1 iret)
 call ncaptc(cdfid,cdzid,'long_name',NCCHAR,44,
1 'differentiation of a model field in vertical',
1 iret)
 call ncaptc(cdfid,cintid,'long_name',NCCHAR,40,
1 'integration of a model field in vertical', iret)
```

```
 call ncaptc(cdfid,csumid,'long_name',NCCHAR,24,
1 'Collocation point values', iret)
 call ncaptc(cdfid,iprmid,'long_name',NCCHAR,23,
1 'mud2 integer parameters', iret)
 call ncaptc(cdfid,fprmid,'long_name',NCCHAR,21,
1 'mud2 float parameters', iret)
 call ncaptc(cdfid,mgoptid,'long_name',NCCHAR,32,
1 'initialization options in "init"', iret)
 call ncaptc(cdfid,xminid,'long_name',NCCHAR,17,
1 'minimum longitude', iret)
 call ncaptc(cdfid,yminid,'long_name',NCCHAR,16,
1 'minimum latitude', iret)
 call ncaptc(cdfid,xmaxid,'long_name',NCCHAR,17,
1 'maximum longitude', iret)
 call ncaptc(cdfid,ymaxid,'long_name',NCCHAR,16,
1 'maximum latitude', iret)
 call ncaptc(cdfid,sclmid,'long_name',NCCHAR,20,
1 'salinity climatology', iret)
 call ncaptc(cdfid,tclmid,'long_name',NCCHAR,23,
1 'temperature climatology', iret)
 call ncaptc(cdfid,uclmid,'long_name',NCCHAR,43,
1 'velocity in longitude direction climatology', iret)
 call ncaptc(cdfid,vclmid,'long_name',NCCHAR,42,
1 'velocity in latitude direction climatology', iret)
 call ncaptc(cdfid,dtid,'long_name',NCCHAR,13,
1 'time interval', iret)
 call ncaptc(cdfid,timeid,'long_name',NCCHAR,4,
1 'time', iret)
```

```
* leaves define mode
      call ncendf(cdfid, iret)

* initializes variables
      do 30 i = 1,lp
        do 20 j = 1,mp
          do 10 k = 1,np
            xiu(i,j,k) = 0.0
            xiv(i,j,k) = 0.0
            xit(i,j,k) = 0.0
            xis(i,j,k) = 0.0
10        continue
          psi(i,j) = 0.0
          h(i,j) = 0.0
          f(i,j) = 0.0
20      continue
30    continue

      do 40 i = 1,charlen
        ident(i) = ' '
40    continue

      ll = 0
      ml = 0
      nl = 0
      xl = 0.0
      el = 0.0
```

```
uvnu2 = 0.0

snu2 = 0.0

tnu2 = 0.0

uvnu4 = 0.0

snu4 = 0.0

tnu4 = 0.0

rho0 = 0.0

g = 0.0

rdrg = 0.0

hmin = 0.0

hmax = 0.0

rfac = 0.0

rdmp = 0.0

pm = 0.0

pn = 0.0

dndx = 0.0

dmde = 0.0

dhdx = 0.0

dhde = 0.0

dx = 0.0

de = 0.0

xp = 0.0

yp = 0.0

pi = 0.0

sig = 0.0

cp = 0.0

cf = 0.0

cd = 0.0
```

```
      cdz = 0.0
      cint = 0.0
      csum = 0.0
      iprm = 0
      fprm = 0
      mgopt = 0
      xmin = 0.0
      ymin = 0.0
      xmax = 0.0
      ymax = 0.0
      sclm = 0.0
      tclm = 0.0
      uclm = 0.0
      vclm = 0.0
      dt = 0.0
      time = 0.0

* writes values into netCDF variables
      call ncvpt(cdfid,xiuid,strt3D,cnt3D,xiu,iret)
      call ncvpt(cdfid,xivid,strt3D,cnt3D,xiv,iret)
      call ncvpt(cdfid,xitid,strt3D,cnt3D,xit,iret)
      call ncvpt(cdfid,xisid,strt3D,cnt3D,xis,iret)
      call ncvpt(cdfid,psiid,strt2D,cnt2D,psi,iret)
      call ncvptc(cdfid,identid,1,charlen,ident,charlen,iret)
      call ncvpt1(cdfid,llid,0,ll,iret)
      call ncvpt1(cdfid,mlid,0,ml,iret)
      call ncvpt1(cdfid,nlid,0,nl,iret)
      call ncvpt1(cdfid,xlid,0,xl,iret)
```

```
call ncvpt1(cdfid,elid,0,el,iret)
call ncvpt(cdfid,hid,strt2D,cnt2D,h,iret)
call ncvpt(cdfid,fid,strt2D,cnt2D,f,iret)
call ncvpt1(cdfid,uvnu2id,0,uvnu2,iret)
call ncvpt1(cdfid,snu2id,0,snu2,iret)
call ncvpt1(cdfid,tnu2id,0,tnu2,iret)
call ncvpt1(cdfid,uvnu4id,0,uvnu4,iret)
call ncvpt1(cdfid,snu4id,0,snu4,iret)
call ncvpt1(cdfid,tnu4id,0,tnu4,iret)
call ncvpt1(cdfid,rho0id,0,rho0,iret)
call ncvpt1(cdfid,gid,0,g,iret)
call ncvpt1(cdfid,rdrgid,0,rdrg,iret)
call ncvpt1(cdfid,hminid,0,hmin,iret)
call ncvpt1(cdfid,hmaxid,0,hmax,iret)
call ncvpt1(cdfid,rfacid,0,rfac,iret)
call ncvpt1(cdfid,rdmpid,0,rdmp,iret)
call ncvpt1(cdfid,pmid,0,pm,iret)
call ncvpt1(cdfid,pn,0,pn,iret)
call ncvpt1(cdfid,dndxid,0,dndx,iret)
call ncvpt1(cdfid,dmdeid,0,dmde,iret)
call ncvpt1(cdfid,dxid,0,dx,iret)
call ncvpt1(cdfid,deid,0,deid,iret)
call ncvpt1(cdfid,xpid,0,xp,iret)
call ncvpt1(cdfid,ypid,0,yp,iret)
call ncvpt1(cdfid,piid,0,pi,iret)
call ncvpt1(cdfid,sigid,0,sig,iret)
call ncvpt1(cdfid,cpid,0,cp,iret)
call ncvpt1(cdfid,cfid,0,cf,iret)
```

```
      call ncvpt1(cdfid,cdid,0,cd,iret)
      call ncvpt1(cdfid,cdzid,0,cdz,iret)
      call ncvpt1(cdfid,cintid,0,cint,iret)
      call ncvpt1(cdfid,csumid,0,csum,iret)
      call ncvpt1(cdfid,iprmid,0,iprm,iret)
      call ncvpt1(cdfid,fprmid,0,fprm,iret)
      call ncvpt1(cdfid,mgoptid,0,mgopt,iret)
      call ncvpt1(cdfid,xminid,0,xmin,iret)
      call ncvpt1(cdfid,xmaxid,0,xmax,iret)
      call ncvpt1(cdfid,ymaxid,0,ymax,iret)
      call ncvpt1(cdfid,sclmid,0,sclm,iret)
      call ncvpt1(cdfid,tclmid,0,tclm,iret)
      call ncvpt1(cdfid,uclmid,0,uclm,iret)
      call ncvpt1(cdfid,vclmid,0,vclm,iret)
      call ncvpt1(cdfid,dtid,0,dt,iret)
      call ncvpt1(cdfid,timeid,0,time,iret)

* closes netCDF file
      call ncclos(cdfid, iret)

* end of program
      end
```

The corresponding CDL is:

```
netcdf hist {
dimensions:
nhist = 4 ;
```

```
charlen = 36 ;
num_long = 130 ;
num_lat = 130 ;
num_levels = 5 ;

variables:
float xiu(num_levels, num_lat, num_long) ;
xiu:long_name = "velocity in longitude dimension" ;
float xiv(num_levels, num_lat, num_long) ;
xiv:long_name = "velocity in latitude dimension" ;
float xit(num_levels, num_lat, num_long) ;
xit:long_name = "temperature" ;
float xis(num_levels, num_lat, num_long) ;
xis:long_name = "temperature" ;
float psi(num_lat, num_long) ;
psi:long_name = "horizontal transport stream function" ;
char ident(charlen) ;
ident:long_name = "run identification (YYMMDD,type of data)" ;
float xl ;
xl:long_name = "longitude interval length" ;
float el ;
el:long_name = "latitude interval length" ;
float h(num_lat, num_long) ;
h:long_name = "bottom depth model" ;
float f(num_lat, num_long) ;
f:long_name = "coriollis force" ;
float uvnu2 ;
uvnu2:long_name = "laPlacian coefficient of friction velocity" ;
```

```
float snu2 ;
snu2:long_name = "laPlacian coefficient of friction salinity" ;
float tnu2 ;
tnu2:long_name = "laPlacian coefficient of friction temperature" ;
float uvnu4 ;
uvnu4:long_name = "biharmonic coefficient of friction velocity" ;
float snu4 ;
snu4:long_name = "biharmonic coefficient of friction salinity" ;
float tnu4 ;
tnu4:long_name = "biharmonic coefficient of friction temperature" ;
float rho0 ;
rho0:long_name = "constant density" ;
float g ;
g:long_name = "gravity" ;
float rdrg ;
rdrg:long_name = "bottom drag coefficient" ;
float hmin ;
hmin:long_name = "minimum bottom depth" ;
float hmax ;
hmax:long_name = "maximum bottom depth" ;
float rfac ;
rfac:long_name = "simulation conversion factor (.5 g/p.)" ;
float rdmp ;
rdmp:long_name = "weights for nudging forcing" ;
float pm ;
pm:long_name = "Jacobian of grid transformation in lon dimension" ;
float pn ;
pn:long_name = "Jacobian of grid transformation in lat dimension" ;
```

```
float dndx ;
dndx:long_name = "Hermetian of \"pn\" with respect to longitude" ;
float dmde ;
dmde:long_name = "Hermetian of \"pm\" with respect to latitude" ;
float dhdx ;
dhdx:long_name = "derivating of topography with respect to longitude" ;
float dhde ;
dhde:long_name = "derivating of topography with respect to latitude" ;
float dx ;
dx:long_name = "interval in longitude dimension" ;
float de ;
de:long_name = "interval in latitude dimension" ;
float xp ;
xp:long_name = "longitude grid points" ;
float yp ;
yp:long_name = "latitude grid points" ;
float pi ;
pi:long_name = "pi" ;
float sig ;
sig:long_name = "stretched coordinate collocation points" ;
float cp ;
cp:long_name = "polynomial basis set" ;
float cf ;
cf:long_name = "matrix of basis functions at the sigma locations" ;
float cd ;
cd:long_name = "derivative of basis with respect to vertical" ;
float cdz ;
cdz:long_name = "differentiation of a model field in vertical" ;
```

```
float cint ;
cint:long_name = "integration of a model field in vertical" ;
float csum ;
csum:long_name = "Collocation point values" ;
long iprm ;
iprm:long_name = "mud2 integer parameters" ;
long fprm ;
fprm:long_name = "mud2 float parameters" ;
long mgopt ;
mgopt:long_name = "initialization options in \"init\"" ;
float xmin ;
xmin:long_name = "minimum longitude" ;
float ymin ;
ymin:long_name = "minimum latitude" ;
float xmax ;
xmax:long_name = "maximum longitude" ;
float ymax ;
ymax:long_name = "maximum latitude" ;
float sclm ;
sclm:long_name = "salinity climatology" ;
float tclm ;
tclm:long_name = "temperature climatology" ;
float uclm ;
uclm:long_name = "velocity in longitude direction climatology" ;
float vclm ;
vclm:long_name = "velocity in latitude direction climatology" ;
float dt ;
dt:long_name = "time interval" ;
```

```
float time ;
time:long_name = "time" ;
}
```

To move back and forth between the CDL and the program, the user moves to the location of the netCDF utilities. There they execute the ncgen utility with options to create a FORTRAN program that reads the file and a netCDF file itself (if desired). The utility call

```
ncgen -f -o <filename.write>
```

generates the corresponding FORTRAN program. A C language option is also available.

8.2 INOCONV

The utility INOCONV (INO Conversion) can now be run on this FORTRAN program that writes the netCDF file to generate a FORTRAN program that reads the same netCDF file. This is executed using the command line:

```
inoconv < routine.write.f > routine.read.f
```

The utility program is straightforward and is presented here in its entirety.

```
c

      program pttogt
c fortran header
c       module name: pttogt (change netcdf writes (PuTs) TO reads (GeTs))
c       module type: program system utility
cmodule environment: kept in ecmop directory for general use
cmodule description: maps ncvp to ncvg calls in netcdf and comments out
c                    the define state
cparameters - input: none
```

```
c             output: none
c          called by: none
c               call: none
c       commons used: none
cfiles       - input: *.in
c             output: *.out
c fortran header
       character*1 linein(80),lineout(80)
       logical mcont
       open(18,file='text.in',status='old')
       open(17,file='text.out',status='unknown')
       mcont=.false.
       do 100 i=1,1000
         read(18,10,end=9999) linein
 10      format(80A1)
         do 50 j=1,80
         lineout(j)=linein(j)
         if ((linein(j).eq."n").and.(linein(j+1).eq."c")) then
c
c  looking for nccre ncvp or ncendf
c
         if (mcont) then
c
c  if we are in define mode, we are looking only for ncendf
c
          if ((linein(j+2).eq."e").and.(linein(j+3).eq."n").and.
     x         (linein(j+4).eq."d").and.(linein(j+5).eq."f")) then
            mcont=.false.
```

```
          lineout(1)="c"
         endif
        else
         if ((linein(j+2).eq."c").and.(linein(j+3).eq."r").and.
     x        (linein(j+4).eq."e")) then
c
c  we have found the entrance to define mode, start commenting out
c  the lines
c
          mcont=.true.
         endif
         if ((linein(j+2).eq."v").and.(linein(j+3).eq."p")) then
           lineout(j+3)="g"
         endif
        endif
        endif
 50     continue
       if (mcont) lineout(1)="c"
       write(17,10) (lineout(ii),ii=1,80)
 100   continue
       goto 9999
 9990  continue
 9999  continue
       if (mcont) then
c
c  error, should not terminate still in define mode
c
       write(6,11)
```

```
11    format('terminated in define mode, error in input file')
      endif
      stop
      end
c
```

8.3 Design Constraints

If the netCDF jackets overhead exceeds 20 percent we may wish to reconsider our design. Ideally we hope not to exceed 5 percent on average.

9 Validation Criteria

This section discusses how we will know that the netCDF data dictionary and utilities and subroutines are performing correctly. It will be the basis for the system tests.

9.1 Performance Bounds

The netCDF interfaces must add no more than twenty percent overhead to the model run, and we would like to see no more than five percent.

An associated software report routine will be called to document each netCDF activity to disk. This report will document that the communications that we expected to take place did in fact occur.

9.2 Test Cases

Our timing test case will run the models with and without the netCDF interface and time the run costs of each to determine the percentage of overhead. This will be done averaged over five separate overall tests to average out system load affects, and over five separate cases for each module interface.

The software results should be equivalent in all respects, except performance, up to the level of precision used by the netCDF file format.

10 Bibliography

[1] Hedstrom, K.S., User's Manual for a Semi-Spectral Primitive Equation Regional Ocean-circulation Model, Ver.3.0beta, INO, May 19, 1990.

[2] Mellor, G.L., Documentation for a 3-D, Primitive Equation, Numerical Ocean Model, Princeton University, Rev. June 15, 1989.

[3] NEPRF Data Dictionary, Monterey CA

[4] O'Connor, Bill, A Users Manual For The Princeton Gulf Stream Model (PRIGS), Aug. 1990.

[5] Perkins, A. Louise, Data Dictionary for the INO netCDF Interfaces, 1990.

[6] Perkins, A. Louise, Specifications For The Data Assimilation (ASSM) Module Version 1.0, 1990.

[7] Rew, R.K., netCDF User's Guide An Interface for Data Access, Unidata, Ver. 1.0, April 1989.

DISTRIBUTION LIST

1. Office of Naval Research
Code 1242 (10 copies)
800 North Quincy Street
Arlington, VA 22217-5000

2. Director, Atmospheric Sciences
Directorate
NOARL West (Code 400)
Monterey, CA 93943-5000

3. Commanding Officer
Fleet Numerical Oceanography Office
Monterey, CA 93943-5000

4. Commanding Officer
Naval Oceanographic Office
Stennis Space Center, MS 39529

5. Technical Director
CNOC (Code OOT)
Stennis Space Center, MS 39529

6. Commanding Officer
NOARL (Code 100)
Stennis Space Center, MS 39529

7. Technical Director
NOARL (Code 110)
Stennis Space Center, MS 39529

8. Director, Ocean Sciences
Directorate
NOARL (Code 300)
Stennis Space Center, MS 39529

9. Head, Ocean Sensing and
Prediction Division
NOARL (Code 320)
Stennis Space Center, MS 39529

10. Library (3 copies)
NOARL (Code 125)
Stennis Space Center, MS 39529

11. Dr. J. Dana Thompson
NOARL, Code 320
Building 1103, Room 248
Stennis Space Center, MS 39529

12. Dr. William J. Schmitz, Jr.
Department of Oceanography
Woods Hole Oceanographic Inst.
Woods Hole, MA 02543

13. Dr. William Holland
National Center for Atmospheric
Research
P. O. Box 3000
Boulder, CO 80307

14. UCAR Library
P. O. Box 3000
Boulder, CO 80307

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. Agency Use Only *(Leave blank).*

2. Report Date.
October 1991

3. Report Type and Dates Covered.
Technical Memo

4. Title and Subtitle.
Software Requirements Document for the netCDF Interface Within ECMOP

5. Funding Numbers.
Program Element No. 61153N
Project No. R310300
Task No. 801
Accession No. DN250022

6. Author(s).
Louise Perkins
Wen Qian

7. Performing Organization Name(s) and Address(es).
Institute for Naval Oceanography, Building 1103, Room 233, Stennis Space Center, MS 39529

8. Performing Organization Report Number.
TM-3

9. Sponsoring/Monitoring Agency Name(s) and Address(es).
Naval Oceanographic and Atmospheric Research Laboratory
Stennis Space Center, MS 39529

10. Sponsoring/Monitoring Agency Report Number.

11. Supplementary Notes.

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited.
Institute for Naval Oceanography, Stennis Space Center, MS 39529-5005

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*
This document presents the software requirements for the Network Common Data Format (netCDF) interfaces that are being used to modularize the Experimental Center for Mesoscale Ocean Prediction (ECMOP) being built at the Institute for Naval Oceanography (INO). ECMOP is currently under development. It has been designed to be a bread-board test bed for ocean model evaluation and comparisons.

14. Subject Terms.
(U) INO (U) NAOPS (U) SPEM (U) PRINCETON (U) NOGUFS
(U) MODEL (U) PREDICTION (U) HARVARD (U) DART

15. Number of Pages.
68

16. Price Code.

17. Security Classification of Report.
Unclassified

18. Security Classification of This Page.
Unclassified

19. Security Classification of Abstract.
Unclassified

20. Limitation of Abstract.
SAR